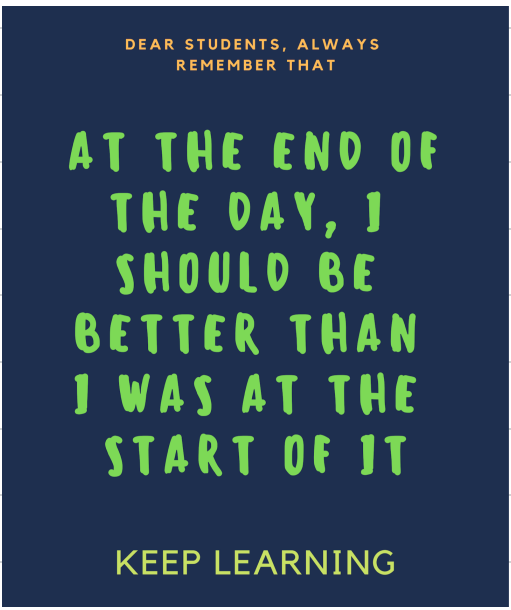
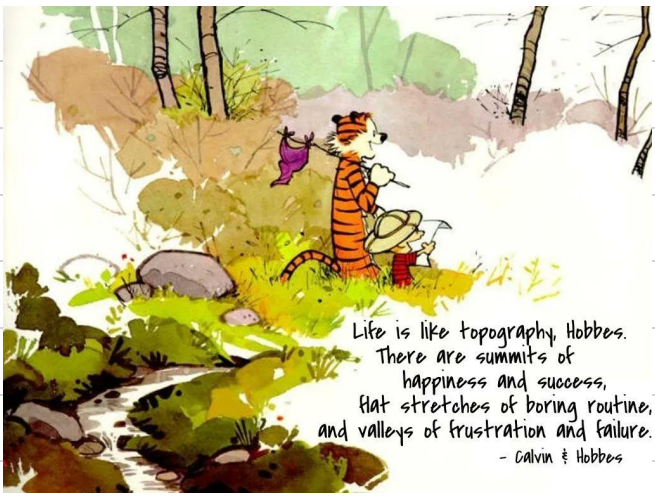
Session 10

Apache Spark

Nov 03, 2025





Issue with Pandas??

— > 100 GB

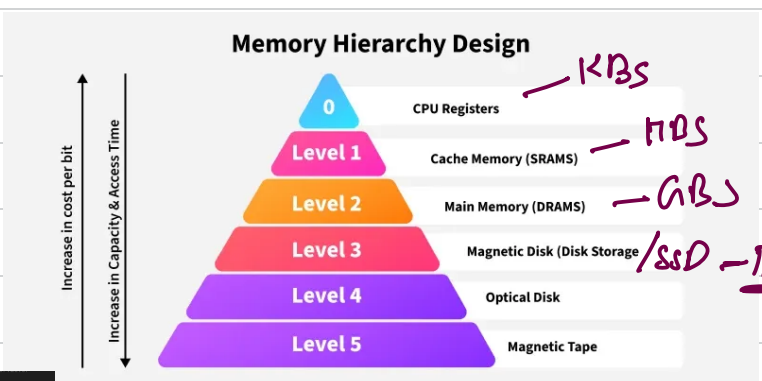
1. It tries to load everything in Ram - single node
2. Single Core
3. Does not use distributed computing.

```
id,name,age,salary
1,Alice,29,100000
2,Bob,31,95000
....
```

) → Pandas into RAM

operational + structural overhead.

RAM - DDR5
DDR4
DDR3



Here's how cache is laid out on the Ryzen 9 7950X3D:

1. **L1 cache (per core, private)**
   - 32 KB instruction + 32 KB data = **64 KB per core**
   - 16 cores → about **1 MB total L1** (but it's per-core, not shared).
   - (Standard Zen 4 layout.)
2. **L2 cache (per core, private)**
   - **1 MB L2 per core** → 16 cores = **16 MB L2 total**.
   - This 16 MB is the "+16 MB" you see in some spec sheets.
3. **L3 cache (per CCD, shared) — and this is the special part**
   - The 7950X3D has **two 8-core chiplets (CCDs)**.
   - **CCD #1 (the V-Cache CCD):** normal 32 MB L3 + stacked 64 MB 3D V-Cache = **96 MB L3** for those 8 cores.
   - **CCD #2 (the regular CCD):** just the normal **32 MB L3** for its 8 cores.
   - So **total L3 = 96 MB + 32 MB = 128 MB**.
   - This is why AMD markets the chip as having **"144 MB cache" = 128 MB L3 + 16 MB L2**. L1 is not counted.

100 chunks, where each chunk contains 10,000 rows using Pandas like this:

```python
import pandas as pd

# Load a large CSV file in chunks of 10,000 rows
for chunk in pd.read_csv('large_file.csv', chunksize=10000):
    print(chunk.shape) # process the shape of each chunk
```

Output:

```python
import pandas as pd

# Load a large CSV file in chunks of 10,000 rows
for chunk in pd.read_csv('large_file.csv', chunksize=10000):
    print(chunk.shape)
```

```
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
(10000, 9)
```

https://api.proxpc.com/media/uploads/2025/01/28/vram-deepseek-table-1.png

Model Variant	Parameters	VRAM (FP16)	VRAM (4-bit Quantization)
DeepSeek-LLM 7B	7 billion	16 GB	4 GB
DeepSeek-LLM 67B	67 billion	154 GB	38 GB
DeepSeek V2 16B	16 billion	37 GB	9 GB
DeepSeek V2 236B	236 billion	543 GB	136 GB
DeepSeek V2.5 236B	236 billion	543 GB	136 GB
DeepSeek V3 671B	671 billion	1,543 GB	386 GB

Dask

70% of API similar to Pandas

1. Dask chunk wise processing

100 GB CSV fi.

Load up 16 GB

↓

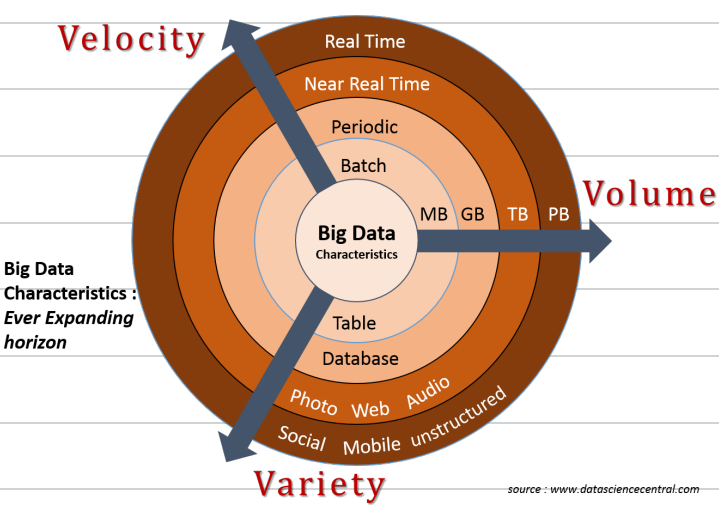
Process

store

2. Lazy Evaluation

3. Distributed Computing (not very stable)

BIG-DATA

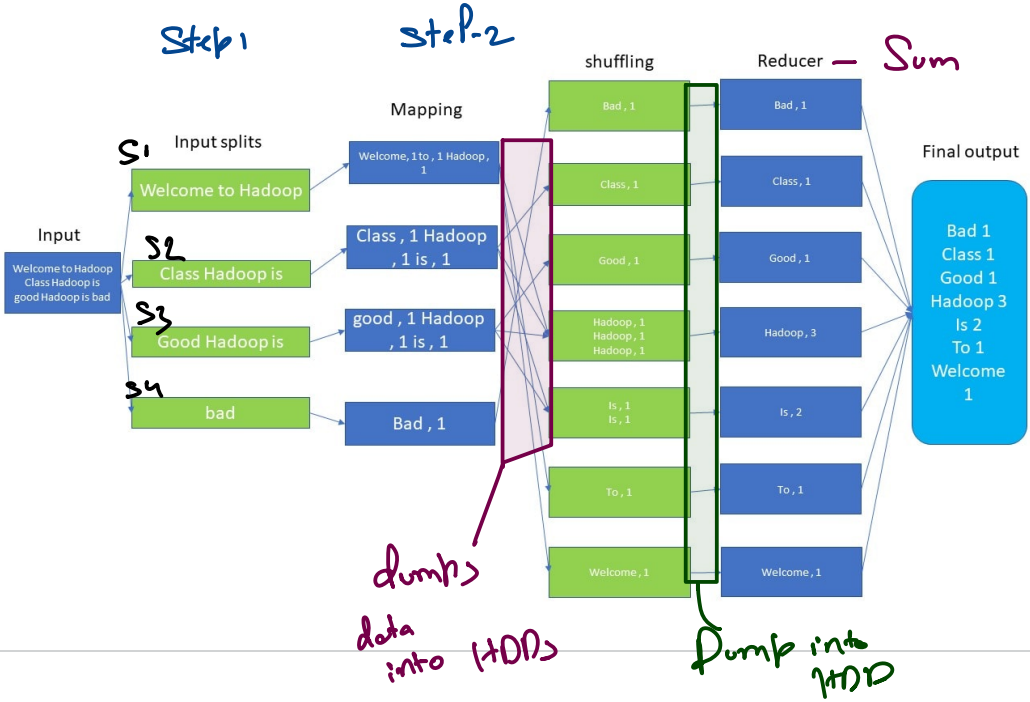


Big Data Characteristics : *Ever Expanding horizon*

source : www.datasciencecentral.com

Google processes approximately **9.5 million** searches every minute as of 2025 data.

Map reduce framework



Nov 3 2025 - 07:51:02

The MapReduce framework was first published by Google in the paper **"MapReduce: Simplified Data Processing on Large Clusters"** by Jeffrey Dean and Sanjay Ghemawat, presented at **OSDI in December 2004**.

For context: the idea became public in 2004 → then Doug Cutting & team used the paper to build the open-source implementation that became **Hadoop MapReduce** starting around **2005–2006**.

Inverted Index

'the dog'

the → 1, 2, 3, 4, 5

dog → 2, 3 — 2, 3

index	word	doc_id	doc_ids
0	and	5,7,9,10	5,7,9,10
1	are	10	10
2	big	6,6	6
3	blue	4	4
4	bright	4	4
5	brown	1	1
6	clever	5	5
7	data	6,7	6,7
8	dog	2,3	2,3
9	fox	1,5	1,5
10	fun	8,9	8,9
11	great	10	10
12	in	4	4
13	is	5,8	5,8
14	jump	3	3
15	jumps	2	2
16	lazy	2,3	2,3
17	learning	7,8	7,8
18	machine	7,8	7,8
19	means	6	6
20	never	3	3
21	numpy	9,10	9,10
22	over	2,3	2,3
23	pandas	9,10	9,10
24	power	6	6
25	quick	1,5	1,5
26	quickly	3	3
27	science	7	7
28	sky	4	4
29	sun	4	4
30	the	1,2,3,4,5	1,2,3,4,5
31	with	9	9

Apache Spark

1. Keeps everything in RAM – distributed way
2. Open Source
3. Distributed Computing
4. Lazy Evaluation

Groceries

1. Milk
2. Bread
3. 2 kg Potatoes
4. 1 kg Rice

Eager Execution

1. Home → Store (Milk) → Home
2. Home → Store (Bread) → Home
3. Home → Onion (Potato) → Home

Lazy Evaluation.

1. In the end, I buy the stuff.

c/c++ / Java → code

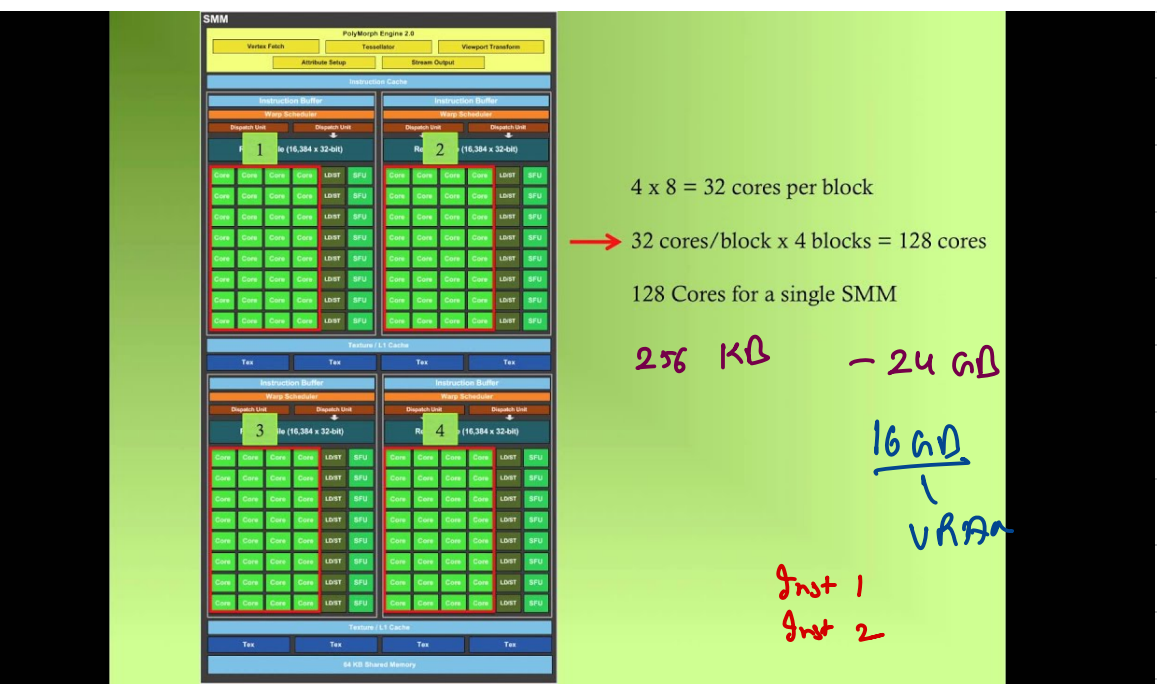
gcc some_file.c

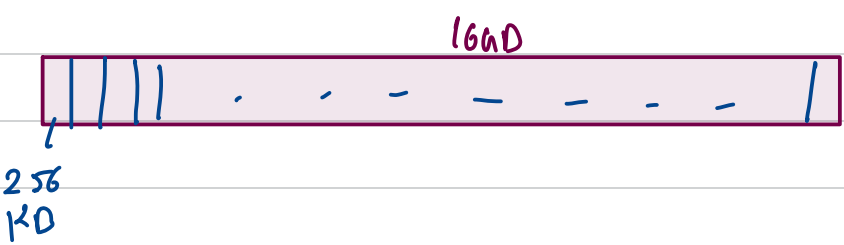
↓

zombie → machine for

↓

Executable





GPU → VRAM – 24GB

CORE – 256KB

1. Divide data of 16 GB into blocks of 256KB
2. Bring each block one by one, into Core (of size 256KB) for execution, post execution dump it back into VRAM.
3. Do this, one by one, for all the blocks.

What is a key limitation of using pandas for Big Data processing that Apache Spark addresses more effectively?

0 users have participated

A	Pandas has more complex functions for simple data analysis tasks.	0%
✓ B	Pandas operates primarily on in-memory data and may struggle with datasets that exceed available RAM.	0%
C	Pandas has a more extensive library for machine learning algorithms compared to Spark.	0%
D	Pandas can only process data in structured formats like CSV or Excel.	0%

End Quiz Now

based on all quizzes from the session

2	Anoop	1/1 ⚡ 96.37
1	Rajesh Sharma	1/1 ⚡ 97.83
3	Aditya Pokhriyal	1/1 ⚡ 95.93
4	Parnapalli Siva Kumar	1/1 ⚡ 91.13
5	Dharani kondlapudi	1/1 ⚡ 90.30
6	Surya Prakash	1/1 ⚡ 89.04
7	sarita Chauhan`	1/1 ⚡ 87.70
8	Riddhi Tatke	1/1 ⚡ 87.53
9	Vinay Yelanadu	1/1 ⚡ 86.30
10	Manoj Narware	1/1 ⚡ 83.30

What is the primary advantage of using Apache Spark over traditional data processing tools for handling Big Data?

0 users have participated

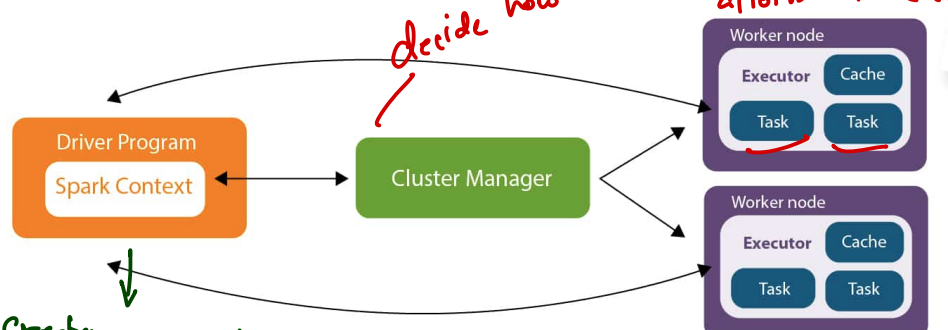
A	Spark can only handle structured data types	0%
B	Spark requires a deep understanding of Java programming.	0%
C	Spark restricts the use of machine learning libraries.	0%
✓ D	Spark provides fast data processing for large, complex datasets.	0%

End Quiz Now

2	Aditya Pokhriyal	2/2 ⚡ 191.67
1	Rajesh Sharma	2/2 ⚡ 193.63
3	Anoop	2/2 ⚡ 190.50
4	Parnapalli Siva Kumar	2/2 ⚡ 188.93
5	Dharani kondlapudi	2/2 ⚡ 184.46
6	Riddhi Tatke	2/2 ⚡ 180.90
7	sarita Chauhan`	2/2 ⚡ 177.07
8	Surya Prakash	2/2 ⚡ 176.70
9	Vinay Yelanadu	2/2 ⚡ 175.10
10	Manoj Narware	2/2 ⚡ 171.13

RAM / CPU should be allotted to each task

decide how much



Creates & assigns data partitions to slave

How does PySpark handle data partitioning by default when creating an RDD?

0 users have participated

A	PySpark keeps all data on a single partition to enhance data security.	0%
B	PySpark does not support partitioning; all data must be managed manually.	0%
C	By automatically determining the number of partitions based on available resources and datasets.	0%
D	PySpark randomly scatters data across the network without regard for resource optimization.	0%

End Quiz Now

Leaderboard

Based on all quizzes from the session

2	Aditya Pokhriyal	3/3 286.20
1	Rajesh Sharma	3/3 290.43
3	Anoop	3/3 286.13

4	Dharani kondlapudi	3/3	276.30
5	Surya Prakash	3/3	267.93
6	sarita Chauhan`	3/3	262.60
7	Manoj Narware	3/3	261.43
8	Parnapalli Siva Kumar	3/3	260.07
9	Riddhi Tatke	3/3	254.96
10	Saurabh Arunkumar Gupta	3/3	254.59